**SQL Assignment 2**

1. For an online purchasing database, create entity relationship diagrams. Create a database object from your entity diagram.

Username Password email ID P\_ID P\_Price P\_name

Customer Online Shopping Products

Buy

Payment Cart Card Items

Payment\_id Amount Payment\_type Crat\_ID Quantity Total\_cost

1. Create a SQL store process to register the use of the database, complete it with proper validation and transaction rollback and commit.

A stored procedure is a prepared SQL code that you can save, so the code can be reused over and over again. So if you have an SQL query that you write over and over again, save it as a stored procedure, and then just call it to execute it. You can also pass parameters to a stored procedure, so that the stored procedure can act based on the parameter value(s) that is passed.

1. List the SQL aggregate function and demonstrate how to utilize it.

AVG – calculates the average of a set of values.

COUNT – counts rows in a specified table or view.

MIN – gets the minimum value in a set of values.

MAX – gets the maximum value in a set of values.

SUM – calculates the sum of values.

Create database dsp

Use dsp

Select \* from customer (create customer table)

select sum(age) from customer

select avg(age) from customer

select min(age) from customer

select max(age) from customer

1. In SQL, create a pivot query.

Pivot in SQL is used when we want to transfer data from row level to column level and Unpivot in SQL is used when we want to convert data from column level to row level. PIVOT and UNPIVOT relational operators are used to generate a multidimensional reporting.

First, select a base dataset for pivoting.

Second, create a temporary result by using a derived table or common table expression (CTE)

Third, apply the PIVOT operator.

1. With an example, describe how to join in SQL.

A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

create table customer

(

cust\_id int not null,

cust\_name varchar(50),

contact\_name varchar(50),

address varchar(50),

postalcode varchar(50),

country varchar(10),

PRIMARY KEY(cust\_id)

)

select \* from customer

create table cust\_orders

(order\_id int not null,

cust\_id int ,

emp\_id int,

order\_date datetime,

shipping\_Id varchar(20),

primary key(order\_id)

)

drop table cust\_orders

select \* from cust\_orders

insert into customer values(1,'abhi','abhi','mumbai','00000','india');

insert into customer values(2,'ak','ak','dhili','00001','india');

insert into customer values(3,'thor','thor','nanded','000012','india');

insert into customer values(4,'sank','sank','panjab','00013','india');

insert into customer values(5,'niks','niks','hariyana','00004','india');

insert into customer values(6,'kish','kish','gujrat','00005','india');

insert into customer values(7,'bag','bag','him','00006','india');

insert into cust\_orders values(2,2,002,sysdate()-10,101);

insert into cust\_orders values(5,5,003,sysdate()-20,102);

insert into cust\_orders values(3,3,004,sysdate()-30,103);

insert into cust\_orders values(4,4,005,sysdate()-25,101);

select \* from customer

Query for left join:-

select \* from customer c left join cust\_orders o

on c.cust\_id=o.cust\_id

Query for Right join:-

select \* from customer c right join cust\_orders o

on c.cust\_id=o.cust\_id

Query for Inner join:-

select \* from customer c inner join cust\_orders o

on c.cust\_id=o.cust\_id

Query for Full outer join:-

select \* from customer c left outer join cust\_orders o

on c.cust\_id=o.cust\_id

union

select \* from customer c right outer join cust\_orders o

on c.cust\_id=o.cust\_id

1. How to locate the 4th highest value in a column in a row. Create your table.

create table employee

(

emp\_ID int,

emp\_salary int

)

insert into employee values (1,50000);

insert into employee values (2,45000);

insert into employee values (3,60000);

insert into employee values (4,15000);

insert into employee values (5,850000);

insert into employee values (6,45000);

insert into employee values (7,35000);

insert into employee values (8,800000);

insert into employee values (9,10000);

select \* from employee

select \* from employee

order by emp\_salary desc

limit 1

offset 3